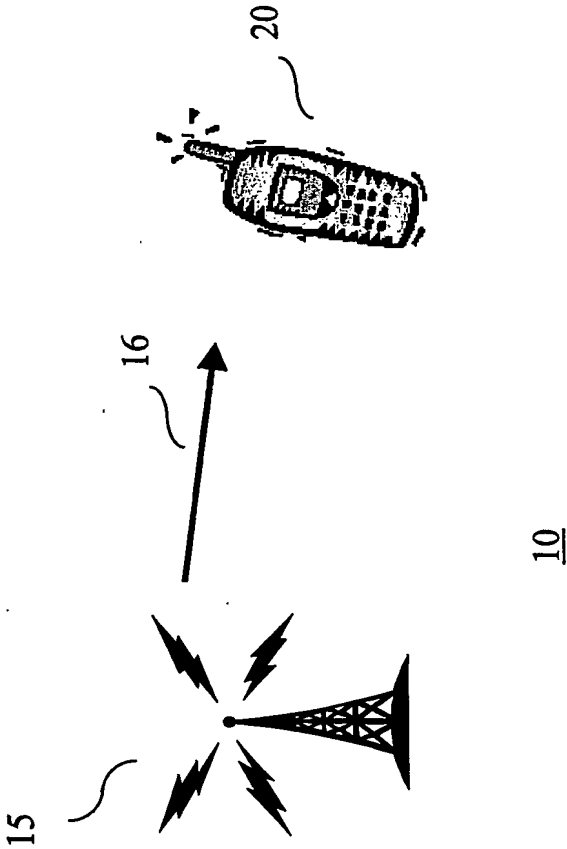


FIG. 1



2/14

FIG. 2

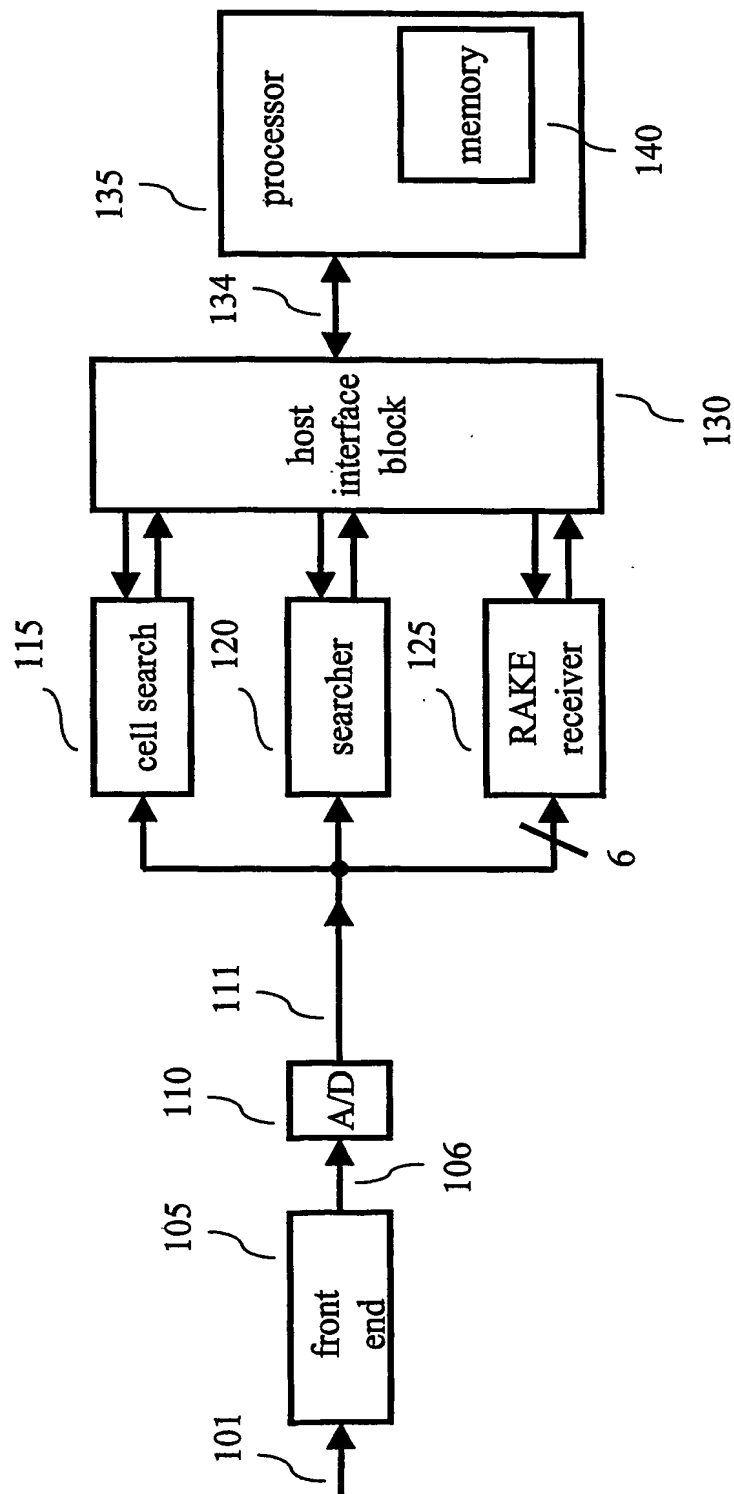
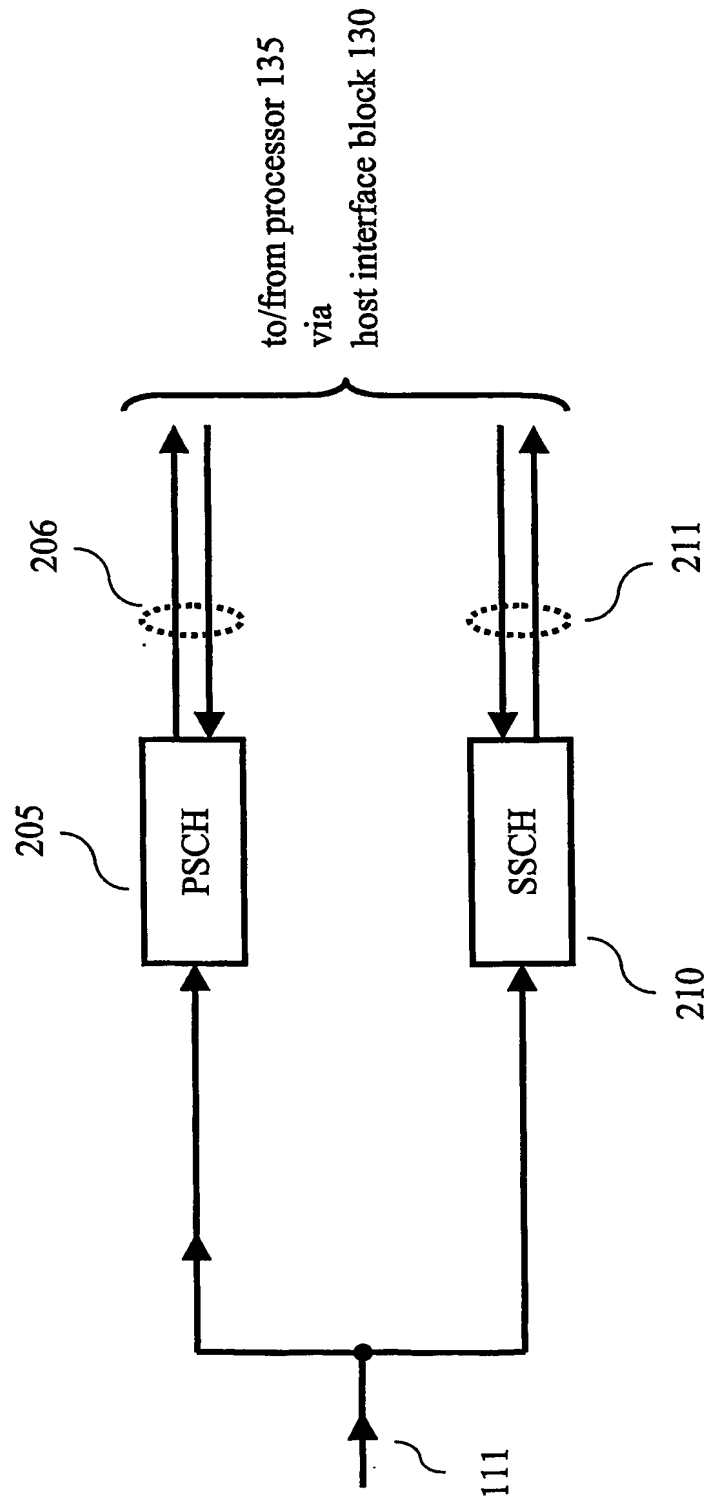
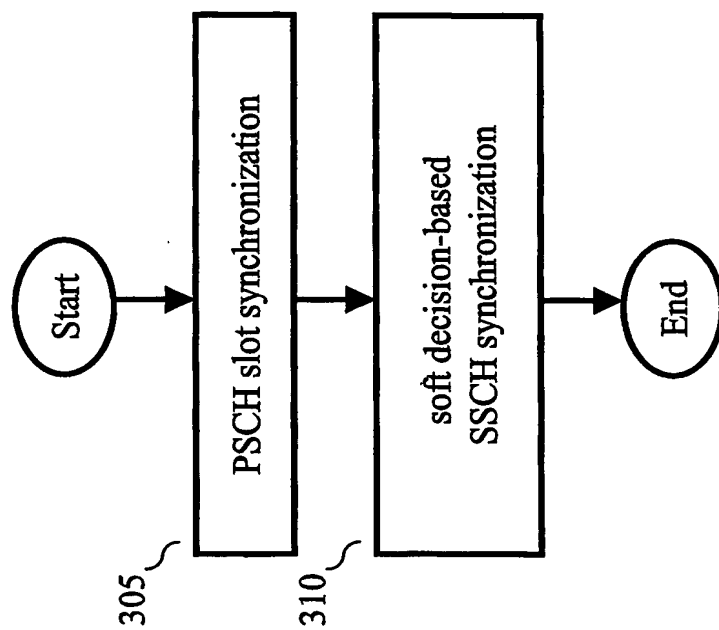


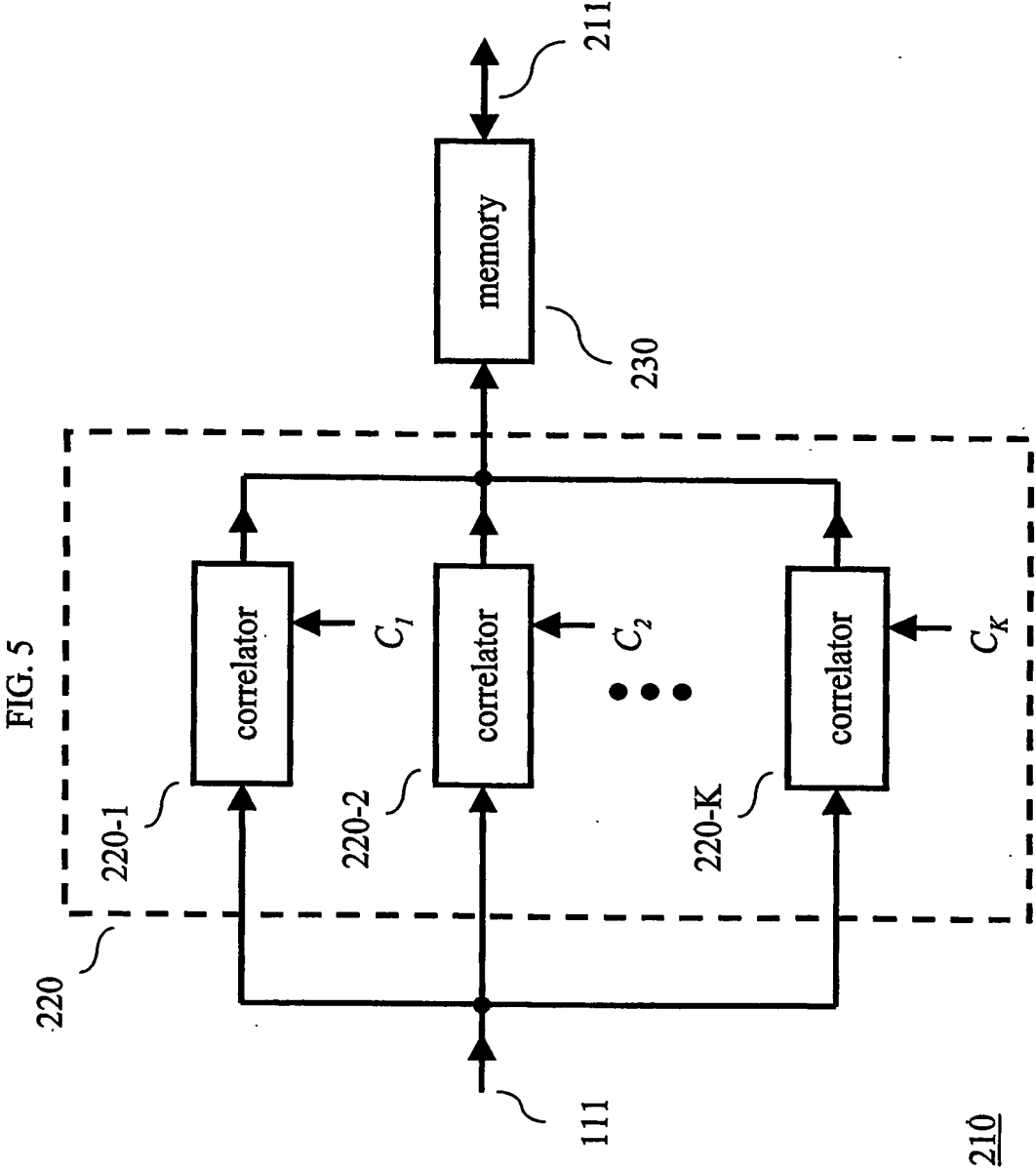
FIG. 3



4/14

FIG. 4





6/14

FIG. 6
Matrix of Correlator Values

[illegible]

7/14

FIG. 7

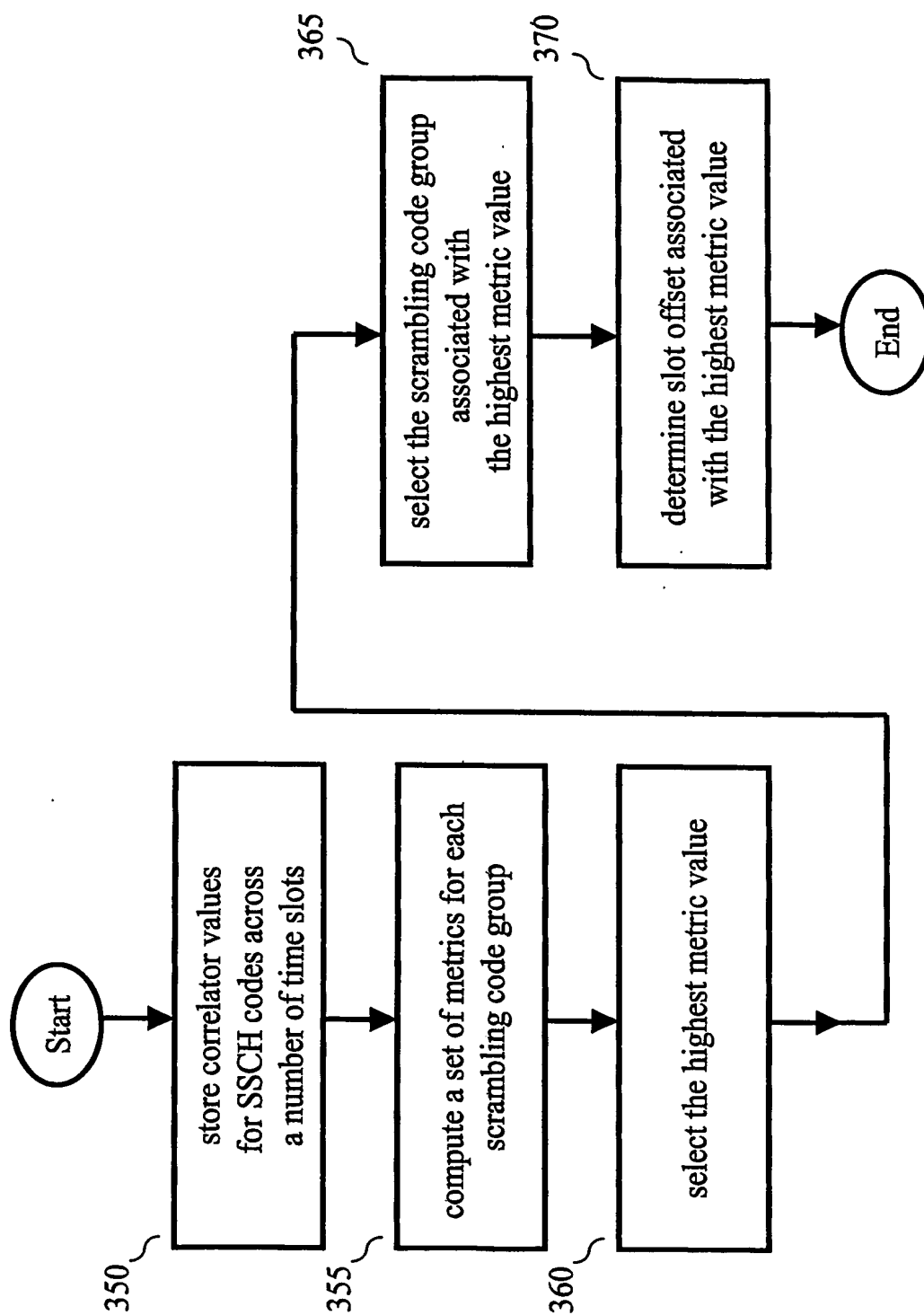


FIG. 8

213

Metric Matrix

Shift

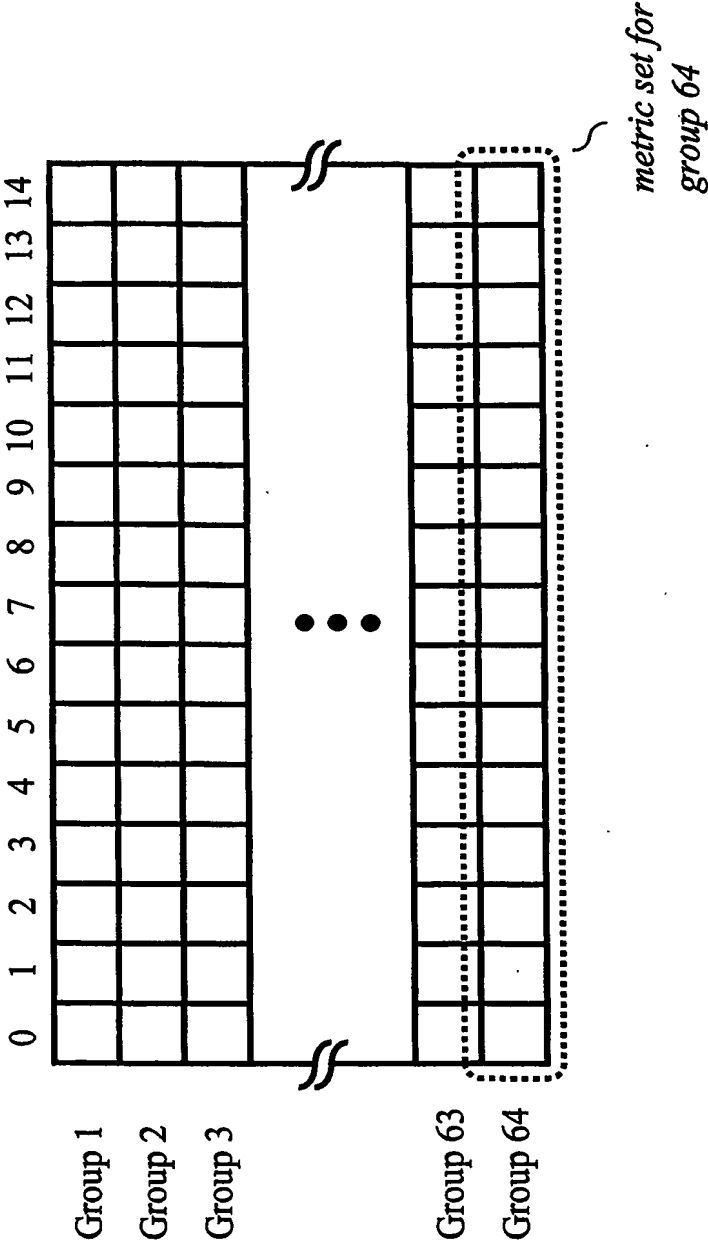


FIG. 9
Matrix of Correlator Values

Code Alphabet = {1, 2, 3, 4}

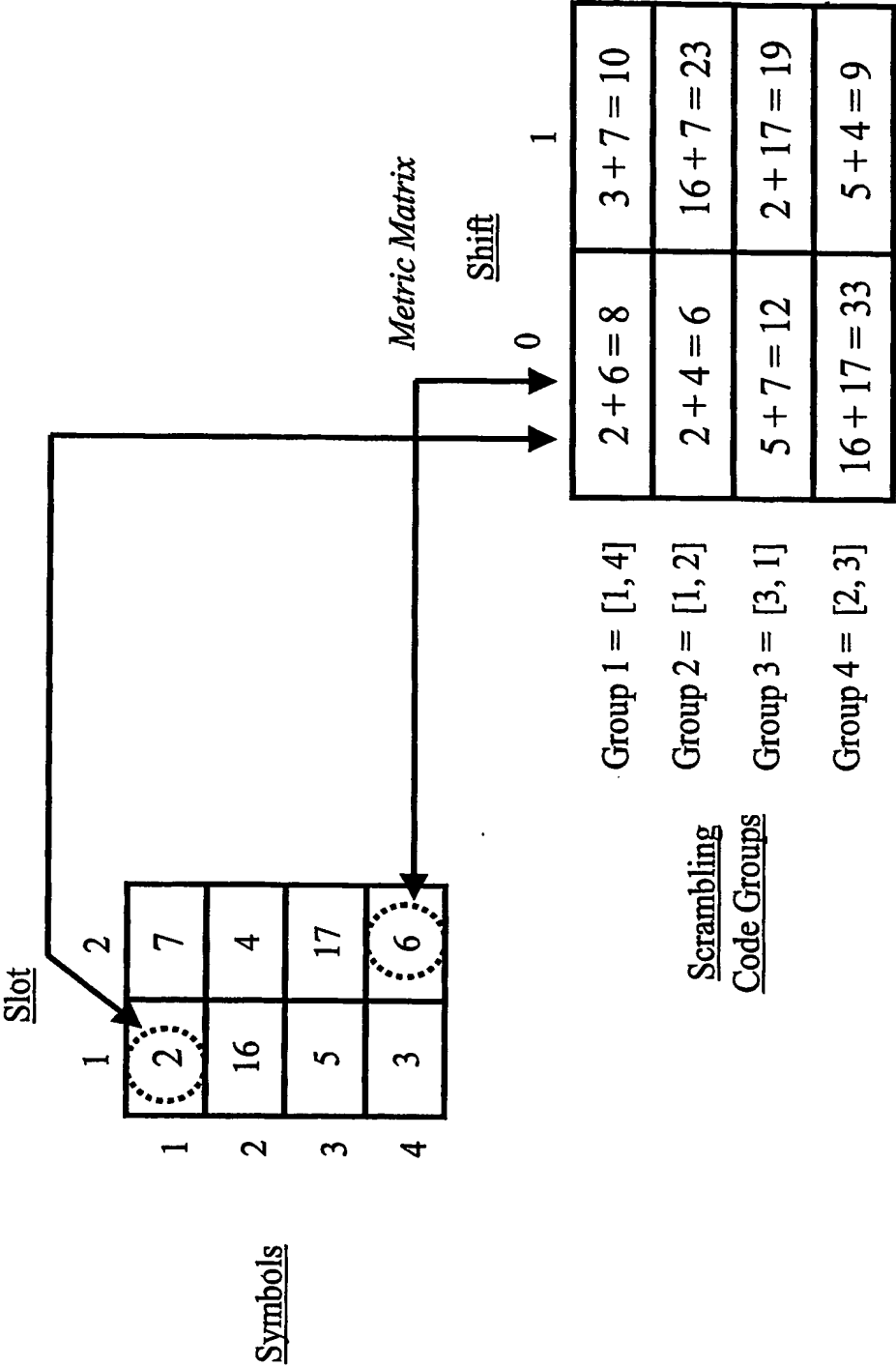
	<u>Slot</u>	
	1	2
<u>Symbols</u>	2	7
1		
2	16	4
3	5	17
4	3	6

FIG. 10

<u>Scrambling</u> <u>Code Groups</u>		<i>Metric Matrix</i>	
		<u>Shift</u>	
		0	1
Group 1 = [1, 4]	Group 1	2 + 6 = 8	3 + 7 = 10
Group 2 = [1, 2]	Group 2	2 + 4 = 6	16 + 7 = 23
Group 3 = [3, 1]	Group 3	5 + 7 = 12	2 + 17 = 19
Group 4 = [2, 3]	Group 4	16 + 17 = 33	5 + 4 = 9

FIG. 11

Matrix of Correlator Values



12/14

FIG. 12

```

// initialize variables
confidence_bins[16,15] = 0
metric [64,15] = 0

// compute confidence values from 16 parallel correlations
for slot_cnt = 1 to 15
    // 15 slots
    for i = 1 to 16
        // 16 parallel correlations
        confidence_bins(i, slot_cnt) = confidence_bins(i, slot_cnt) + abs(rx_data[1:256] * SSC[i,1:256])
    end
end

// compute metrics to find match with highest confidence
for code_group = 1 to 64
    for shift = 1 to 15
        for slot = 1 to 15
            val = confidence_bins[group_seq[code_group, mod(slot + shift, 15)], slot]
            metric[code_group, shift] = metric[code_group, shift] + val
        end
    end
end

// find max value in metrics matrix
max_val = 0
for code_group = 1 to 64
    for shift = 1 to 15
        if (metric[code_group, shift] > max_val)
            max_val = metric[code_group, shift]
            group = code_group
            offset = shift
        end
    end
end

```

FIG. 13

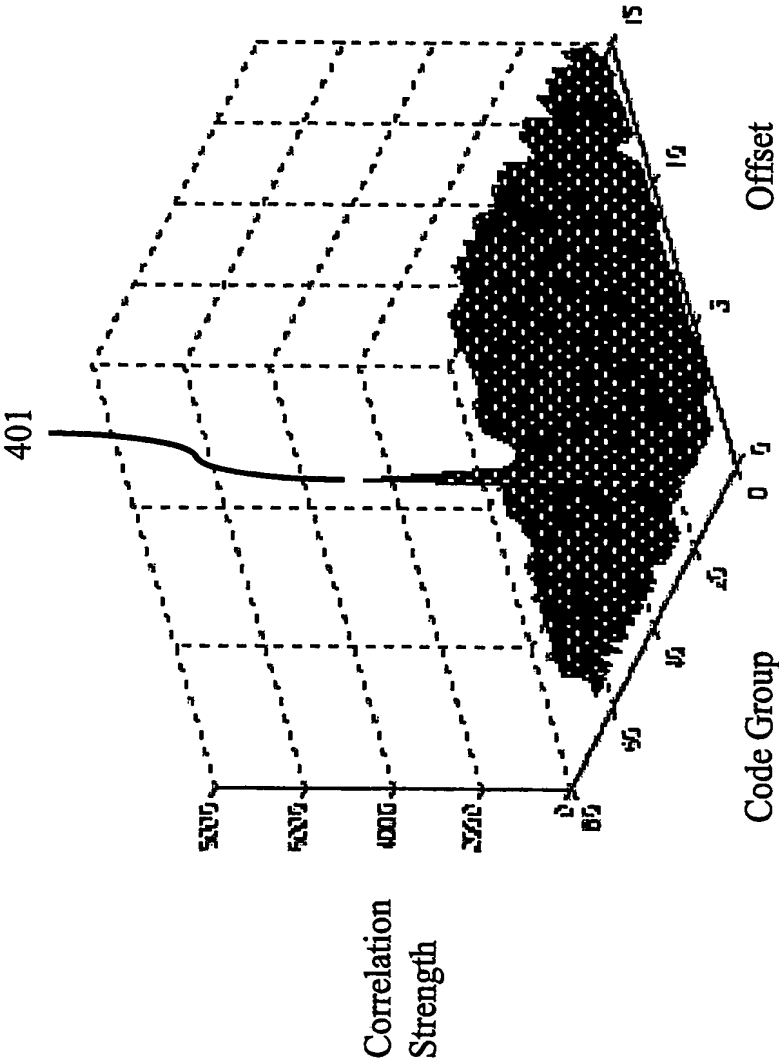


FIG. 14

